Amendment to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A computer program product residing on a computer readable storage medium and comprising instructions, including a branch instruction that when executed causes a data processing apparatus to:

cause an executing instruction stream to branch to an instruction at a specified address if a state, of a state name specified in the branch instruction is a specified value, with the state indicating the availability of a resource of the data processing apparatus, wherein the state is available to a plurality of microengines, each of the plurality microengines having control logic and an execution box, the execution box including an arithmetic logic unit and a general purpose register set, each of the plurality of microengines being configured to process a plurality of threads.

2. (Previously presented) The computer program product of claim 1 wherein the state is set to a logic one or a logic zero by a processor.

3. (Previously presented) The computer program product of claim 1 wherein the state is set to logic one or a logic zero by a microengine in a parallel processor.

4. (Cancelled)

5. (Previously presented) The computer program product of claim 1 further comprising:

an optional token that is set by a programmer and specifies a number i of instructions to execute following the branch before performing the branch operation where the number of instructions can be specified as one, two or three.

6. (Cancelled)

7. (Previously presented) The computer program product of claim 1 wherein causing the executing instruction stream to branch causes a branch if the value of the specified state name is set to a logic one.

8. (Previously presented) The computer program product of claim 1 wherein causing the executing instruction stream to branch causes a branch if the value of the specified state name is cleared to a logic zero.

9. (Previously presented) The computer program product of claim 1 wherein the state name is the name assigned to an executing context.

10. (Currently amended) A method of operating a processor having multiple microengines comprises:

evaluating a value of a state name specified in a branch instruction, the value of the state name indicating the availability of a resource of the processor; and

performing a branching operation based on the value of the specified state name being set or cleared,

wherein the state is available to the multiple microengines, each of the multiple microengines having control logic and an execution box, the execution box including an arithmetic logic unit and a general purpose register set, each of the plurality of microengines being configured to process a plurality of threads.

11. (Currently amended) The method of claim 10 wherein the state is set to a logic one or a logic zero by [[[a]] the processor.

12. (Currently amended) The method of claim 11 wherein the state is set to logic one or a logic zero by ~~a microengine~~ one of the multiple microengines in ~~a parallel~~ the processor.

13. (Original) The method of claim 11 further comprising:

branching to an instruction at a branch target field specified as a label in the instruction.

14. (Original) The method of claim 11 further comprising:

executing a number i of instructions following execution of the branch instruction before performing the branch operation based on evaluating an optional token that is set by a programmer.

15. (Original) The method of claim 11 further comprising:

evaluating an optional token that is set by a programmer; and

executing one instruction following execution of the branch instruction before performing the branch operation based on whether the optional token is set.

16. (Currently amended) A parallel processor comprises:

a plurality of microengines, each comprising:

a register stack; and

an arithmetic logic unit coupled to the register stack and a program control store that stores a branch instruction that causes the ~~processor~~ microengine to:

evaluate a value of a state name specified in the branch instruction, the value of the state name indicating an availability of a resource of the processor; and

perform a branching operation based on the value of the specified state name being set or cleared,

wherein the state is available to the plurality of microengines,.

17. (Currently amended) The processor of claim 16 ~~wherein the processor is a parallel processor with the register stack an arithmetic logic unit is part of a microengine, and the processor further comprises:~~

~~at least an additional microengine, the microengine comprising:~~

~~a register stack;~~

~~an arithmetic logic unit coupled to the register stack and a program control store,~~ ~~and~~ wherein the state is set to logic one or a logic zero by one of the microengines.

18. (Original) The processor of claim 16 further comprising:

a branch target field specified as a label in the instruction.

19. (Currently amended) A computer program product residing on a computer readable storage medium comprising instructions, including a branch instruction that when executed causes a processor to:

evaluate a value of a state name specified in the branch instruction, the value of the specified state name indicating an availability of a resource of the processor; and

perform a branching operation based on the value of the specified state name being set or cleared,

wherein the state is available to a plurality of microengines, each of the plurality microengines having control logic and an execution box, the execution box including an arithmetic logic unit and a general purpose register set, each of the plurality of microengines being configured to process a plurality of threads.

20. (Original) The computer program product of claim 19 wherein instructions to perform a branching operation branch if the value of the specified state name is set to a logic one.

21. (Previously presented) The computer program product of claim 1, wherein the resource of the data processing apparatus includes a queue.

22. (Previously presented) The computer program product of claim 21, wherein the queue is a FIFO request receive queue.